Feature Selection for Data Mining

H. Liu and L. Yu
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-5406
hliu@asu.edu, leiyu@asu.edu

Abstract

This survey is a comprehensive overview of many existing feature selection algorithms from the 1970's to the present. In this survey, we categorize different algorithms under a general framework based on different search strategies and evaluation criteria, and reveal hitherto unattempted combinations of search strategies and evaluation criteria. We propose a unifying platform, on which all existing algorithms can be compared and evaluated on an equal-footing, which serves an important intermediate step towards automatic feature selection. Guidelines for applying the unifying platform are illustrated through representative algorithms. In this survey, we also identify future research areas in feature selection, introduce newcomers to this field, and pave the way for automatic selection of feature selection algorithms, which will greatly facilitate practitioners in searching for suitable algorithms for solving domain specific real-world applications.

Key words: Feature Selection, Framework, Platform

1 Introduction

Data mining [AIS93, HF96, PK99, LLFH99], as a multidisciplinary joint effort from databases, machine learning, and statistics, is championing in turning mountains of data into nuggets. Feature selection plays an important role in data selection and preparation for data mining [FPSS96, KJ97, LM98a, Pyl99]. It reduces the dimensionality of feature space, removes redundant, irrelevant, or noisy data. It brings the immediate effects for application: speeding up a data mining algorithm, improving the data quality and thereof the performance of data mining, and increasing the comprehensibility of the mining results. Feature selection has been a fertile field of research

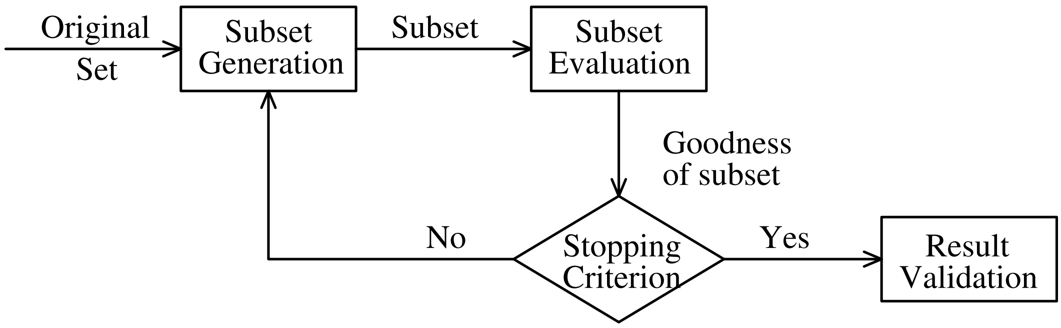

Figure 1: General Precedures of Feature Selection

and development since 1970's in statistical pattern recognition [WDJ80, BB82, SS88, JZ97], machine learning and data mining [BL97, DL97, DB00, KSM00, Das01, MMP02].

Feature selection can be defined as a process that chooses a minimum subset of M features from the original set of N features ($M \leq N$), so that the feature space is optimally reduced according to a certain evaluation criterion. As the dimensionality of a domain expands, the number of feature N increases. Finding the best feature subset is usually intractable [KJ97] and many problem related to feature selection have been shown to be **NP**-hard [BR92]. In general, a feature selection algorithm consists of four basic steps (refer to Figure 1): subset generation, subset evaluation, stopping criterion, and result validation [DL97].

Subset generation is a search procedure. Basically, it generates subsets of features for evaluation. Let N denote the number of features in the original data set, then the total number of candidate subsets is 2^N, which makes exhaustive search through the feature space infeasible with even moderate N. Each subset generated by the generation procedure needs to be evaluated by a certain evaluation criterion and compared with the previous best one with respect to this criterion. If it is found to be better, then it replaces the previous best subset. Without a suitable stopping criterion the feature selection process may run exhaustively before it stops. A feature selection process may stop under one of the following reasonable criteria: (a) a predefined number of features are selected, (b) a predefined number of iterations are reached, (c) whether addition (or deletion) of any feature does not produce a better subset, (d) an optimal subset according to the evaluation criterion is obtained. The selected best feature subset needs to be validated by carrying out different tests on both the selected subset and the original set and comparing the results using artificial data sets and/or real-world data sets.

Researchers have studied various aspects of feature selection. *Search* is a key topic in the study of feature selection [Doa92, Lan94] such as search starting points, search directions, and search strategies. Another important aspect is how to measure the *goodness* of a feature subset [Doa92, LM98c]. There are *filter models* [SS88, FI92, LS96b] and *wrapper models* [JKP94, CF94, DB00] with different emphases on dimensionality reduction or accuracy enhancement. Recently, *hybrid models* are proposed to combine the advantages of both *filter models* and *wrapper models* in the context of handling exponentially high dimensionality [Das01, XJK01]. According to class information availability in data, there are *supervised* feature selection approaches [WK91, DL97] as well as *unsupervised* feature selection approaches [DLY97, DL99, Tal99, DB00].

There have been quite a few attempts to survey the field of feature selection based on some framework or structure. Prominent among these are Siedlecki and Sklansky's [SS88], Doak's [Doa92], Dash and Liu's [DL97], and Blum and Langley's [BL97] surveys. Siedlecki and Sklanksy discussed the evolution of feature selection algorithms and grouped the algorithms into past, present, and future categories. Their main focus was branch and bound algorithm [NF77], and its variants (i.e. [FS87]). No experimental study was conducted in that paper. Doak followed a similar approach to Siedlecki and Sklansky's survey and grouped the different search procedures and evaluation criteria used in feature selection algorithms independently, and run experiments using some combinations of search procedures and evaluation criteria. Dash and Liu's survey summarized the general feature selection procedures into four steps (as shown is Figure 1), and categorized 32 different feature selection algorithms (belonging to either filter models or wrapper models) from 1970's till 1997 into different groups based on the two major steps of feature selection (generation procedure and evaluation criteria). Experiments were conducted to compare and evaluate the strengths and weaknesses of different algorithms and guidelines for selecting feature selection algorithms were given based on data types and domain characteristics. Blum and Langley's survey is similar to Dash and Liu's survey in describing the general procedures of feature selection algorithms and grouping different algorithms. In addition to feature selection, it also covers the topic of selecting relevant instances in machine learning.

In this paper, we follow Dash and Liu's framework in categorizing different feature selection algorithms. However, by covering many recent algorithms, we make the following contributions:

1. adopting latest developments in unsupervised feature selection algorithms into the four-steps general procedures originally designed for supervised feature selection, by adding new evaluation criteria from both filter models and wrapper models into the subset evaluation step,

2. thus extending the framework to cover 45 different algorithms,

3. enriching feature selection models with the newly proposed hybrid model which is a combination of the filter model and wrapper model,

4. based on intensive study of existing algorithms, proposing a unifying platform, which serves: i) to cover all factors in determining the selection of a suitable feature selection algorithm for an application, ii) to provide an equal-footing to compare and evaluate the strengths and weaknesses of all existing algorithms, iii) to categorize existing algorithms from a different point of view (i.e. users or application's view), iv) to pave the way for developing an automatic feature selection system.

5. identifying new research areas, such as feature selection under very high dimensionality and automatic feature selection.

The remainder of this paper is organized as 4 sections. Section 2 gives detailed descriptions of the general feature selection procedures, w.r.t the two major steps, subset generation and subset evaluation, and categorizes 45 different feature selection algorithms based on the different search strategies and evaluation criteria used in them. Section 3 briefly describes the algorithms in each category and discusses the commonalities and differences within each category. Section 4 introduces the development of the unifying platform and illustrates it by applying it on some representative algorithms. Section 5 ends the survey with discussions on refining the unifying platform and further research areas.

2 General Procedures for Feature Selection

In this section, we categorize the two major steps of feature selection procedures: subset generation and subset evaluation. A framework is presented in which a total of 45 algorithms are categorized based on the types of search strategy and evaluation criterion used in them.

2.1 Feature subsets generation

2.1.1 Search direction

First, one must decide the starting point in the search space. One direction is to grow a feature subset from an empty set (i.e. Forward Generation). As the search starts, features are added one at a time. At each iteration, the best feature among unselected ones is chosen based on the evaluation criterion. The subset grows until it reaches a full set of original features or the stopping criterion is reached. An opposite direction is to start from the full set (i.e. Backward Generation). At each iteration, the least important feature is removed based on the evaluation criterion. The subset shrinks until there is only one feature in the set or the stopping criterion is reached. Another way is to start the search from a randomly selected subset (i.e. Random Generation) and add or delete feature at random. There is no particular direction to follow, which tries to avoid being trapped into local optima.

2.1.2 Search strategy

Second, one must decide the search strategy. As we mentioned earlier, an exhaustive search of the space to find the optimal subset is impractical. There are three different strategies to solve this problem, namely: complete, heuristic, non-deterministic [LM98b].

Complete search arises from the answer to the question "Do we have to resort to exhaustive search in order to guarantee the optimality of the resulted subset?". [Sch93] pointed out that in some cases being complete (i.e. no optimal subset is missed) does not necessarily means that search must be exhaustive (i.e. every subset has to be evaluated). If the evaluation criterion possesses some certain property (e.g. monotonicity), we can find an optimal subset without evaluating all of the 2^N subsets. However, this improvement is not enough since the search space is still in the order of $O(2^N)$. More about this is discussed in [NF77].

Heuristic search, as the name suggests, employs heuristics in conducting search. It avoids being complete, but at the same time risks losing optimal subsets. It is sort of depth-first search guided by heuristics. The space complexity could be $O(N^2)$ or less; some exceptions are Relief [KR92] and DTM [Car93] that are discussed in detail in the next section. Algorithms with heuristic search strategy are very simple to implement and very fast in producing results because the search space is only quadratic in terms of the number of features.

Non-deterministic search, unlike the first two types of strategies, searches for the next set at random (i.e. a current set does not directly grow or shrink from any previous set following a deterministic rule). This is also related to anytime algorithm [Zil96], which can generate currently best subset constantly and keep improving the quality of selected subset as time goes by. Although the search space is still $O(2^N)$, but this strategy typically searches fewer number of subsets than 2^N by setting a maximum number of iterations possible. Optimality of the selected subset depends on the resources available.

2.2 Evaluation criteria

An optimal subset is always relative to a certain evaluation criterion (i.e. an optimal subset chosen using one evaluation criterion may not be the same as that using another evaluation criterion). Evaluation criteria can be broadly categorized into two groups based on their dependency on the learning algorithm applied on the selected feature subset. Typically, an *independent criterion* as in filter models tries to evaluate the goodness of a feature or feature subset without the involvement of a learning algorithm in this process. A *dependent criterion* as in wrapper models tries to evaluate the goodness of a feature or feature subset by evaluating the performance of the learning algorithm applied on the selected subset. In other words, it is the same measure on the performance of the applied learning algorithm. In the following two subsections, we briefly discuss each evaluation criterion in these two groups.

2.2.1 Independent criteria

Some of the independent criteria are distance measure, information measure, dependency measure, consistency measure, and similarity measure.

Distance Measures are also known as separability, divergence, or discrimination measures. For a two-class problem, a feature X is preferred to another feature Y if X induces a greater difference between the two-class conditional probabilities than Y; if the difference is zero, then X and Y are indistinguishable. An example is the Euclidean distance measure.

Information measures typically determine the information gain from a feature. The information gain from a feature X is defined as the difference between the prior uncertainty and expected posterior uncertainty using X. Feature X is preferred to feature Y if the information gain from feature X is greater than that from feature Y (e.g. entropy measure) [BB82].

Dependency measures or correlation measures qualify the ability to predict the value of one variable from the value of another. The coefficient is a classical dependency measure and can be used to find the correlation between a feature and a class. If the correlation of feature X with class C is higher than the correlation of feature Y with C, then feature X is preferred to Y. A slight variation of this is to determine the dependence of a feature on other features; this value indicates the degree of redundancy of the feature. All evaluation functions based on dependency measures can be divided between distance and information measures. But, these are still kept as a separate category, because conceptually, they represent a different viewpoint [BB82].

More about the above three types of measures can be found in Ben-Bassets [BB82] survey.

Consistency measures are relatively new and have been in much focus recently. These are characteristically different from other measures, because of their heavy reliance on the training dataset and the use of the Min-Features bias in selecting a subset of features [AD94]. Min-Features bias prefers consistent hypotheses definable over as few features as possible. These measures find out the minimally sized subset that satisfies the acceptable inconsistency rate, that is usually set by the user.

2.2.2 Dependent criteria

For supervised learning, the primary goal of classification is to maximize predictive accuracy, therefore, predictive accuracy is generally accepted and widely used as the primary measure by researchers and practitioners. As the features are selected using the classifier that later on uses these selected features in predicting the class labels of unseen instances, the accuracy level is very high although computationally quite costly [JKP94].

While for unsupervised learning, there exist a number of heuristic criteria for estimating the quality of clustering results, such as cluster compactness, scatter separability, and maximum likelihood.

3 Algorithms Categorized in the Framework

We collect all together 45 feature selection algorithms from the literature and categorize all of them according to the two aspects, search strategies and evaluation criteria. As we can see from the table in the appendix, all of them can be put into 15 different categories, of which category 1 to category 11 belong to supervised feature selection, and the rest four categories belong to unsupervised feature selection. We give brief descriptions of various algorithms in each category in the following subsections.

3.1 Category 1: (Generation - Complete, Evaluation - Distance)

The combination in this category is found in quite old algorithms such as branch and bound [NF77]. Other algorithms in this category are variances of branch and bound algorithm regarding the search strategies or the evaluation criteria used. We first describe the branch and bound algorithm, followed by a brief description of the other two algorithms.

Branch and bound algorithm attempts to satisfy two criteria: (i) the selected subset be as small as possible and (ii) a bound be placed on the value of the evaluation criterion. According to these two criteria, branch and bound algorithm starts searching from the original feature set and proceeds by removing features from it. A bound is placed on the value of the evaluation criterion to make the search fast. As the evaluation criterion obeys the monotonicity principle, any subset for which the value is less than the bound is removed from the search space. The evaluation criterion used is divergence [NF77].

[XYC88] proposes a similar algorithm (BFF), where the search strategy is modified to solve the problem of searching an optimal path in a weighted tree with the informed best first search strategy in artificial intelligence. This algorithm guarantees the globally best subset without exhaustive search, for any criterion satisfying monotonicity principle.

[Bob88] proves that the homogeneity coefficient can be used in measuring the degree of linear dependency among some measurements, and shows that it is applicable to the feature selection problem due to its monotonicity principle. Hence, this can be suitable converted to a feature selection algorithm by implementing it as an evaluation criterion for branch and bound with backtracking or a best first generation procedure.

3.2 Category 2: (Generation - Heuristic, Evaluation - Distance)

As seen in the table, the most prominent algorithm in this category is Relief [KR92]. We first discuss Relief and its variant ReliefF [Kon94], followed by a brief description on the other algorithm.

The key idea of Relief is to estimate the quality of features according to how well their values distinguish between the instances of the same and different classes that are near each other. For this purpose, given a randomly selected instance X from a data set S, with k attributes, Relief searches the data set for its two nearest neighbors: one from the same class, called nearest hit H, and the other from a different class, called nearest miss M. It updates the quality estimation $W[A_i]$ for all the features A_i depending on the difference diff() on their values for X,M, and H. The process is repeated for m times, where m is a user-defined parameter. Time complexity of Relief for a data set with N instances is $O(mkN)$. Clearly, efficiency is one of the major advantages of this algorithm over other algorithms. One major limitation is that it does not help with reducing redundant features. Another limitation is that it works only for binary classes. This limitation is overcome by ReliefF [Kon94] that also tackles incomplete data.

Jakub Segen's [Seg84] algorithm uses an evaluation criterion that minimizes the sum of a statistical discrepancy measure and the feature complexity measure. It finds the first feature that best distinguishes the classes, and iteratively looks for additional features which in combination with the previously chosen features improve class information. This process stops once the minimal representation criterion is achieved.

3.3 Category 3: (Generation - Complete, Evaluation - Information)

Under this category, we have MLDM (Minimum Description Length Method) [SDN90]. In this algorithm, if the features in a subset V can be expressed as a fixed class-independent function F of the features in another subset U (U and V together make up the complete feature set), then once the values of the features in subset U are known, the feature subset V becomes useless. Minimum description length criterion (MDLC) introduced by [J.78] is used in this algorithm. The algorithm exhaustively searches all the possible 2^N subsets and outputs the subset satisfying MDLC. This algorithm can find all useful features and only those if the observations are Gaussian.

3.4 Category 4: (Generation - Heuristic, Evaluation - Information)

In this category, we have three algorithms: SFG, DTM [Car93], and Koller and Sahami's [KS96].

In SFG (Sequential Forward Generation), the algorithm starts with an empty set S, and adds features from the original set F sequentially, i.e. one by one. At each round of selection, the best feature f is chosen according some information criterion. f is then added into S and removed from F. So, S grows and F shrinks. This algorithm can provide a ranked list of features according to that the first feature chosen is the most relevant and the last feature chosen the least relevant.

The DTM (Decision Tree Method) algorithm adopts the same idea as SFG to generate feature subset, but uses entropy criterion to evaluate each selected subset. In this algorithm, C4.5 [Qui93] is run over the training set, and features that finally appear in the pruned decision tree are selected as the best subset. In other words, the union of the subsets of features, appearing in the path to any leaf node in the pruned tree, is the selected best subset.

The third algorithm is based on the intuition that any feature, given little or no additional information beyond that subsumed by the remaining features, is either irrelevant or redundant, and should be eliminated. To realize this, Koller and Sahami try to approximate *Markov blanket* where a subset T is a *Markov blanket* for feature f_i if, given T, f_i is conditionally independent both of the class label and of all features not in T (including f_i itself). The implementation of *Markov blanket* is sub-optimal in many ways, particularly due to many naive approximations.

3.5 Category 5: (Generation - Heuristic, Evaluation - Dependency)

In this category, we found POE+ACC (Probability of ERROR and Average Correlation Coefficient) algorithm [MG71] and PRESET [Mod93].

As many as seven techniques of feature selection are presented in POE+ACC, but we choose the last (i.e. seventh) method because the authors consider it to be the most important among the seven. In this algorithm, the first feature selected is the feature with the smallest probability of error P_e. The next feature selected is the feature that produces the minimum weighted sum of P_e and average correlation coefficient (ACC), and so on. ACC is the mean of the correlation coefficients of the candidate feature with the features previously selected at that point. This algorithm can rank all the features based on the weighted sum, and a required number of features is used as the stopping criterion.

PRESET uses the concept of rough set; if first finds a reduct (i.e. a reduct R of a set S classifies instances equally well as S does) and removes all features not appearing in the reduct. Then it ranks the features based on their significance. The significance of a feature is a measure expressing how important a feature is regarding classification. This measure is based on dependency of features.

3.6 Category 6: (Generation - Complete, Evaluation - Consistency)

In this category, we have Focus [AD92] and its two variances, and two variances of branch and bound mentioned in category 1.

Focus considers all the combinations among N features starting from an empty set. When Focus finds a subset that satisfies the consistency criterion, it stops. When the number of useful features is not small, the run-time cost is quite prohibitive.

Schimmer's [Sch93] algorithm, and MIFES1 [OV92] algorithm are two variances of Focus. Schlimmer's algorithm uses a systematic enumeration scheme as generation procedure and consistency as the evaluation criterion. It uses a heuristic function that makes the search for the optimal subset faster. This heuristic will be observed is proportional to the percentage of values that have been observed very infrequently considering the subset of features. All supersets of an unreliable subset are also unreliable. MIFES1 is very similar to Focus in its selection of features. It represents the set of instances in the form of matrix, each element of which stands for a unique combination of a positive instance (class=1) and a negative instance (class=0). A feature f is said to cover an element of the matrix if it assumes opposite values for a positive instance and a negative instance associated with the element. It searches for a cover with $N-1$ features starting from one with all N features, and iterates until no further reduction in the size of the cover is attained.

In branch and bound, if consistency is chosen as the evaluation criterion, we can have one variance based on complete generation. Another variance of branch and bound is ABB (Automatic Branch and Bound). By *automatic*, it means that the bound is determined automatically, unlike branch and bound in which the bound is predetermined. In ABB, the algorithm starts with the full set of features, removing one feature at a time in a breadth-first search fashion until no more feature can be removed while the consistency criterion is still satisfied. ABB expands the search space as well as prunes it. A branch is pruned when it cannot grow due to its violation of the inconsistency criterion. When a branch stops growing, its root could be a solution. At the end, a set with the minimum number of features is selected if it satisfies the evaluation criterion.

3.7 Category 7: (Generation - Non-deterministic, Evaluation - Consistency)

In this category, LVF [LS96b] is the representative algorithm, and LVI [LS98] and QBB are two variances of LVF.

LVF starts with a randomly selected subset and randomly searches the space of subsets using a Las Vegas algorithm [BB96] that makes probabilistic choices to help guide them more quickly to an optimal solution. It uses a consistency measure, which attempts to find a minimum number of features that separate classes as consistently as the full set of features can. An inconsistency is defined as two instances having the same feature values but different classes. An inconsistency threshold is fixed in the beginning of the algorithm, and any candidate subset, having an inconsistency rate greater than it, is rejected. The algorithm uses another predefined parameter, the maximum number of subsets randomly generated, as a stopping criterion. An advantage about this algorithm is that the user does not have to wait too long for a good subset because it outputs any subset that is found to be better than the previous best (both by the cardinality of the subset and its inconsistency rate). This algorithm is efficient, as only the subsets whose number of features is smaller or equal to that of the current best subset are checked for inconsistency. One drawback is that it may take more time to find the optimal subset than algorithms using heuristic generation procedures for certain problem, since it cannot take advantage of prior knowledge.

Although LVF can generate optimal/sub-optimal solutions, when data sets are huge, checking whether a data set is consistent takes a lot of time. LVI, an incremental version of LVF, can significantly reduce the number of inconsistency checking by reducing the data used for inconsistency checking while remaining the quality of resulted feature subsets. Another problem about LVF is that although it can produce results very fast in the beginning, it improves very slowly later on. QBB algorithm solves this problem by combining LVF and ABB (in category 6). This is because ABB applies backward generation, which makes it suitable when there is only a small number of irrelevant features in the original feature set, and LVF can efficiently remove some irrelevant features in the initial stage of search, and when ABB takes over the search, the number of irrelevant features is smaller than the one in the beginning of the search. However, it should be noted that QBB cannot guarantee the optimality of the final selected subset because the output of LVF may not be the superset of optimal subset.

3.8 Catergory 8: (Generation - Complete, Evaluation - Classifier Error Rate)

In this category under complete search strategy, we found four wrapper algorithms. Ichino and Sklanksy have devised two algorithms based on two different classifiers, namely: linear classifier [IS84a] and box classifier [IS84b]. In both of the two algorithms, they solve the feature selection problem using zero-one integer programming [Geo67]. AMB and B (approximate Monotonic Branch and Bound) was introduced to combat the disadvantage of branch and bound by permitting evaluation criteria that are not monotonic [FS87]. In this algorithm, the bound of branch and bound is relaxed to generate subsets that appear below some subset violating the bound; but the selected subset should not violate the bound. BS (Beam Search) [Doa92] algorithm is a type of best-first search that uses a bounded queue to limit the scope of the search. The queue is ordered from best to worst with the best subsets at the front of the queue. The generation procedure proceeds by taking the subset at front of the queue, and producing all possible subsets by adding a feature to it, and so on. Each subset is placed in its proper sorted position in the queue. If there is no limit on the size of the queue, BS is an exhaustive search; if the limit on the size of the queue is one, it is equivalent to WSFG (in the next category).

3.9 Category 9: (Generation - Heuristic, Evaluation - Classifier Error Rate)

Under the heuristic search strategy, we find popular wrapper algorithms such as WSFG (Wrapper Sequential Forward Generation) and WSBG (Wrapper Sequential Backward Generation) [DK82]. SBS-SLASH (Sequential Backward Selection-SLASH) [CF94], PQSS ((P,Q) Sequential Search), BDS (Bi-Directional Search) [Doa92], Schemata Search [ML94], RC (Relevance in Context) [Dom97], and Queiros and Gelsema's algorithm [QG84] are variants of WSFG, or WSBG, or both.

WSFG starts from the empty set, and in each iteration generates new subsets by adding a feature selected by some evaluation criterion. WSBG starts from the complete feature set, and in each iteration generates new subsets by discarding a feature selected by some evaluation criterion. SBS-SLASH is based on the observation that when there is a large number of features, some classifiers (e.g., ID3/C4.5) frequently do not use many of them. It starts with the full feature set (like WSBG), but after taking a step, eliminates (slashes) any feature not used in what was learned at that step. BDS allows to search from both ends, whereas PQSS provides some degree of backtracking by allowing both addition and deletion for each subset. If PQSS starts from the empty set, then it adds more features than it discards in each step, and if it starts from the complete feature set, then it discards more features and adds less features in each step. Schemata search starts either from the empty set or the complete set, and in each iteration, finds the best subset by either removing or adding only one feature to the subset. It evaluates each subset using leave-one-out cross validation (LOOCV), and in each iteration, selects the subset having the least LOOCV error. It continues this way until no single-feature change improves it. RC considers the fact that some features will be relevant only in some parts of the space. It is similar to WSBG, but with the crucial difference that it makes local, instance-specific decisions on feature relevance, as opposed to global ones. Queiros and Gelsema's algorithm is similar to WSFG, but it suggests that at each iteration, each feature in various settings by considering different interactions with the set of features previously selected should be evaluated. Two simple settings are: (i) always assume independence of features (do not consider the previously selected features), and (ii) never assume independence (consider the previously selected features). Bayesian classifier error rate as the evaluation criterion was used.

3.10 Category 10: (Generation - Non-deterministic, Evaluation - Classifier Error Rate)

The five algorithms found under non-deterministic search strategy are: LVW [LS96a], GA (Genetic Algorithm) [VI94], SA (Simulated Annealing) [Doa92], RGSS (Random Generation Plus Sequential Selection) [Doa92], and RMHC-PF1 [Ska94].

LVW generates subsets in a perfectly random fashion (it uses Las Vegas algorithm [BB96]). GA and SA, although there is some element of randomness in their generation, follow specific procedures for generating subsets continuously. RGSS injects randomness to WSFG and WSBG. It generates a random subset, and runs WSFG and WSBG starting from the random subset. Random Mutation Hill Climbing-Prototype and Feature Selection (RMHC-PF1) selects prototypes (instances) and features simultaneously for nearest neighbor classification problem, using a bet vector that records both the prototypes and features. It uses the error rate of an 1-nearest neighbor classifier as the evaluation criterion. In each iteration it randomly mutates a bit in the vector to produce the next subset. All these algorithms need proper assignment of values to different parameters. In addition to maximum number of iterations, other parameters may be required, such as for LVW the threshold inconsistency rate; for GA initial population size, crossover rate, and mutation rate; for SA the annealing schedule, number of times to loop, initial temperature, and mutation probability.

3.11 Category 11: (Generation - Heuristic, Evaluation - Information+Accuracy)

This is the last category of supervised feature selection algorithms, in which we found BBHFS (Boosting-Based Hybrid Feature Selection) [Das01] and Xing-etal01's algorithm [XJK01]. Both of them are rather new. They combine the advantages of filter and wrapper algorithms. Like filters, they are very fast and general, while at the same time using knowledge of learning algorithm to inform the search and provide a natural stopping criterion.

In BBHFS, it uses the concept of boosting from computational learning theory, and evaluates the goodness of a feature subset of cardinality K by information gain criterion. Instead of using a pre-defined parameter, number of features to be selected, as stopping criterion, this algorithm uses the training accuracy of a learning algorithm (e.g. Naive Bayes, ID3, k-NN) to provide a natural stopping criterion. The learning algorithm is applied to each newly selected subset with cardinality K, if the training accuracy does not increase compared to the previous selected subset with cardinality $K-1$, the algorithm stops without adding the last feature to the feature subset. Xing-etal01's algorithm is very similar to BBHFS. It also uses information gain to evaluate feature subset, but the features that pass the information gain filter are input to a more computationally intensive subset selection procedure known as Market Blanket Filtering [KS96]. Another difference with BBHFS is that the subsets with different cardinalities are compared using cross-validation, which is more computational intensive compared to using training accuracy.

3.12 Category 12: (Generation - Heuristic, Evaluation - Information)

From this category, we begin to discuss unsupervised feature selection algorithms (i.e. feature selection algorithms for clustering), which are new applications of the general procedures of feature selection we discussed in section 2.

Under this category, we have SBUD (Sequential Backward for Unsupervised Data) algorithm [DLY97]. In this algorithm, one kind of information criterion, entropy measure, is introduced to rank the features and sequential backward search strategy is used to generate different feature subsets from full set. Entropy measure ranks the features in terms of their importance to the concept they represent or to the clustering task. The entropy value of all the instances of the data set ranges from zero to one. It is a measure of the homogeneity of the distribution of the instances. If the instances are more uniformly distributed, the value is more close to one. Otherwise, the value is more close to zero. Removing features from the data set will make the distribution more uniform, thus increase the entropy value. The value will have a less increase, by removing an less important feature than by removing a more important feature. It is shown that this algorithm is able to find important features without class information. The algorithm stops when the predefined number of useful features is reached.

3.13 Category 13: (Generation - Heuristic, Evaluation - Similarity)

In this category, we have Mitr-etal02's algorithm [MMP02]. It uses heuristic search to generate feature subsets and uses a new feature similarity criterion, called maximum information compression index. The superiority of this measure is that it results in a fast algorithm with time complexity $O(M^2N)$, where M is the number of instances and N is the number of features in the data set, which makes it very suitable for data set with very high dimensionality. It also performs well on removing redundant features.

3.14 Category 14: (Generation - Heuristic, Evaluation - Cluster Goodness)

In this category, we describe the unsupervised feature selection algorithms in a wrapper model setting. Rather than wrapping the search for the best feature subset around a supervised learning algorithm, they wrap the search around a clustering algorithm. Under this group, we have Deva-Ram97's algorithm [DR97], FSSEM (Feature Subset Selection and EM clustering) [DB00], and ELSA (Evolutionary Local Search Algorithm) [KSM00].

The wrapper model for unsupervised feature selection is initially proposed in Deva-Ram97's algorithm for hierarchical (conceptual) clustering. They apply sequential forward and backward search to generate candidate feature subsets, and measure the goodness of each subset by measuring the goodness of clusters resulted by applying COBWEB (a hierarchical clustering algorithm) [Fis87] on the training set. To measure the goodness of clusters, they employ an evaluation criterion called *category utility* [GC85]. An interesting aspect of this evaluation criterion is that it blurs the traditional wrapper/filter model distinction - it is like a wrapper model in that the underlying learning algorithm is being used to guide the search but it is like a filter model in that the evaluation criterion measures an intrinsic property of the data rather than some type of predictive accuracy.

In FSSEM, the algorithm applies sequential forward search and wraps feature subset selection around the EM (Expectation-Maximization) clustering algorithm. When the number of clustering k is not known, it wraps around EM-k (EM clustering with order identification), which searches for both k and the clusters. The algorithm uses both the scatter separability criterion and the maximum likelihood criterion to evaluate the goodness of clusters and applies a normalization function to somehow removes the bias towards the dimension of feature space.

ELSA represents the latest development in unsupervised feature selection using wrapper model. Rather than combining the existing evaluation criteria for goodness of clusters, they use ELSA, an evolutionary local selection algorithm that maintains a diverse population of solutions that approximate the Pareto front in a multi-dimensional objective space. K-means and EM are applied to form the clusters based on the selected feature subset.

3.15 Category 15: (Generation - Heuristic, Evaluation - Information+Cluster goodness)

The algorithm found in this category is Dash-Liu00's algorithm [DL00], which is an extension to SBUD (in category 12) by two means. One is that it introduces random sampling to solve the scalability problem of the SBUD algorithm on large data sets. The other is that it adopts the hybrid model and uses clustering algorithm (K-means) to evaluate the ranked features and also uses it to choose the subset that maximizes the clustering quality, which solves the problem of arbitrary selection of optimal feature subset.

4 A Unifying Platform

Although a large number of feature selection algorithms have been developed and still many more are being developed, the current feature selection systems are still very inefficient to use because they involve users with too much of low-level details about the general procedures, and require algorithms designer's expertise to take full advantage of each method. In other words, the fast growth of feature selection research and development is not matched by an equal ability to intelligently select what choices and methods for users facing a complex world. It would be really useful if given an application, the most suitable feature selection method can be automatically recommended. To serve the goal of automatic feature selection, we propose a unifying platform, on which all the existing algorithms can be fairly compared and systematically studied. In section 4.1 we describe a preliminary platform from previous research, in section 4.2 we develop a more general platform based on the previous work, in section 4.3 we wind up building the unifying platform, and in section 4.4 we illustrate the usage of the unifying platform through representative algorithms.

Before that, we first evaluate the necessity of introducing a unifying platform - "could we just compare the algorithms without it?" Assuming we simply conduct a pairwise comparison between existing feature selection algorithms (S in total), we will have a complex situation of $(O(S^2))$ pairwise relationships between the algorithms, which render our comparison very difficult to conduct. Therefore, with many algorithms, we need a unifying platform to lay down the conditions and

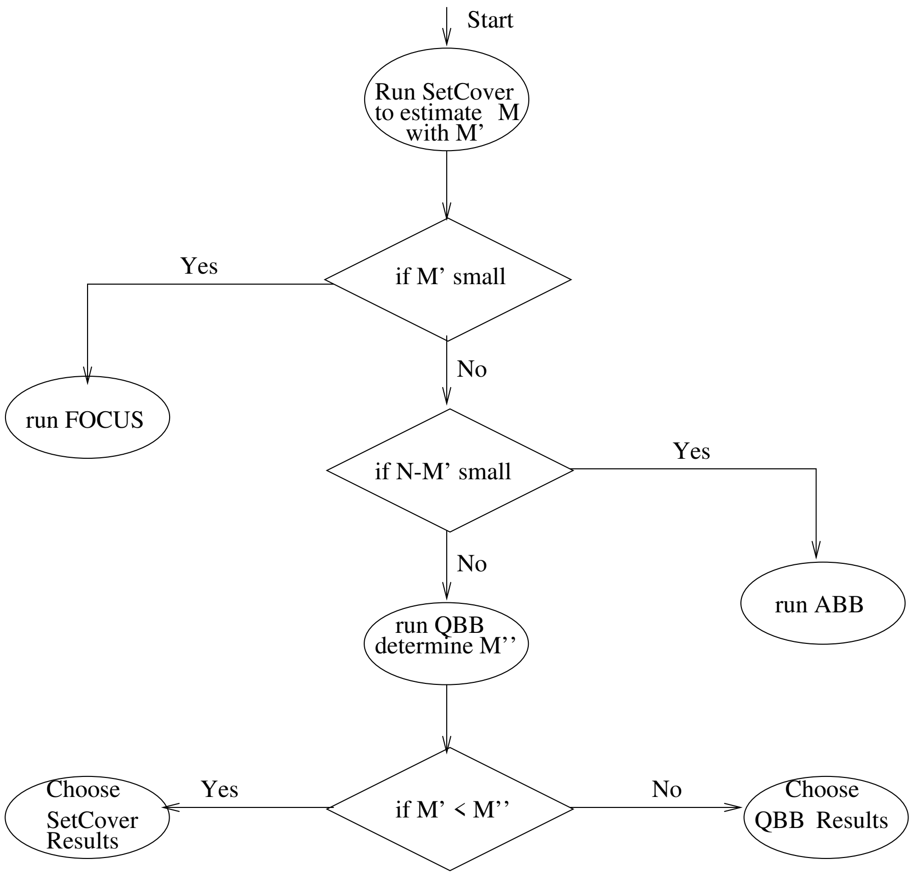

Figure 2: A Preliminary Platform

constrains for a fair and sensible comparison.

4.1 A preliminary platform

In [DLM00], a preliminary platform was developed to automatically choose a suitable feature selection algorithm in a limited scope. The research was focused on only consistency-based feature selection algorithms, including four representative algorithms: Focus - forward exhaustive search, ABB - backward complete search, QBB - hybrid search (random plus ABB), and SetCover - heuristic search. The idea was verified by using the estimated number (M') of features to choose an algorithm shown in Figure 2, where M' is an estimate of M by SetCover, M'' an estimate of M by QBB, and M the number of optimal features that should be selected. Focus and ABB should be used whenever possible as they optimize the subset of selected features under the conditions when

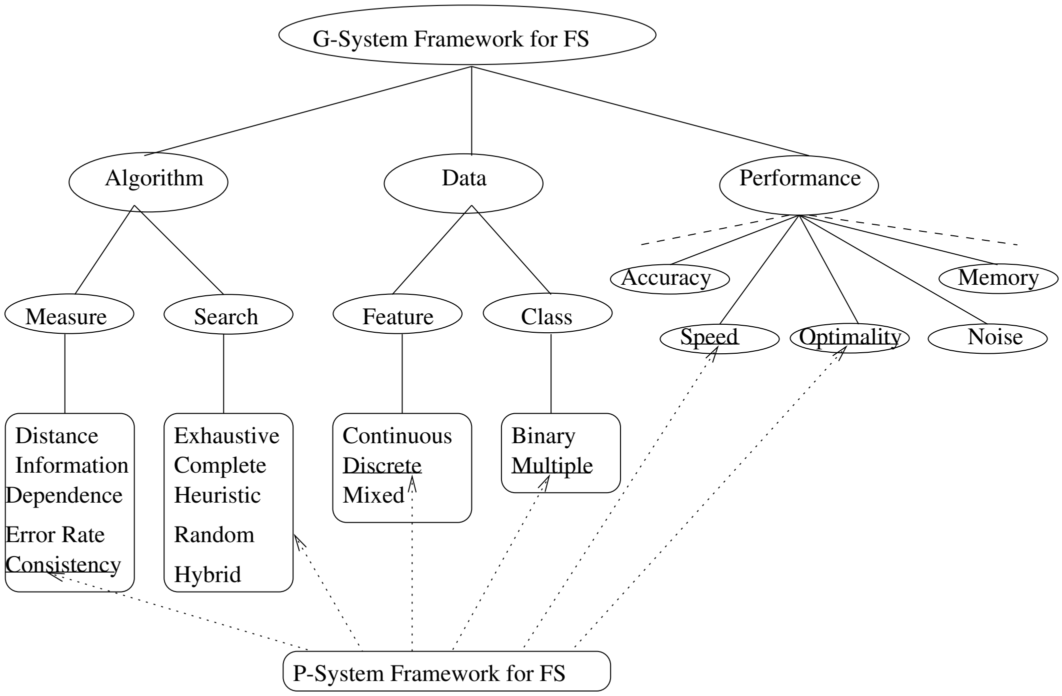

Figure 3: A More General Platform

M or $N - M$ is small. Otherwise, they could take impractically long time to converge. Time factor is implied in the use of the estimator of M.

4.2 A more general platform

The success of the preliminary platform leads us to developing a more general platform that automatically chooses a feature selection algorithm according to some data characteristics assuming one wants to obtain an efficient yet optimal feature subset. It is clear in Figure 3 that the more general platform we developed includes the preliminary platform. In Figure 3, how to choose a feature selection algorithm is determined by three major factors: *algorithm*, used to select features in terms of evaluation measures and search strategies, *data* available, and *performance* required. Each of the three interacted factors has their own dimensions as shown in the figure.

4.3 A unifying platform

In this section, we propose a unifying platform in Figure 4 based on the above two platforms and the recent development of feature selection algorithms. At the top, knowledge and data about fea-

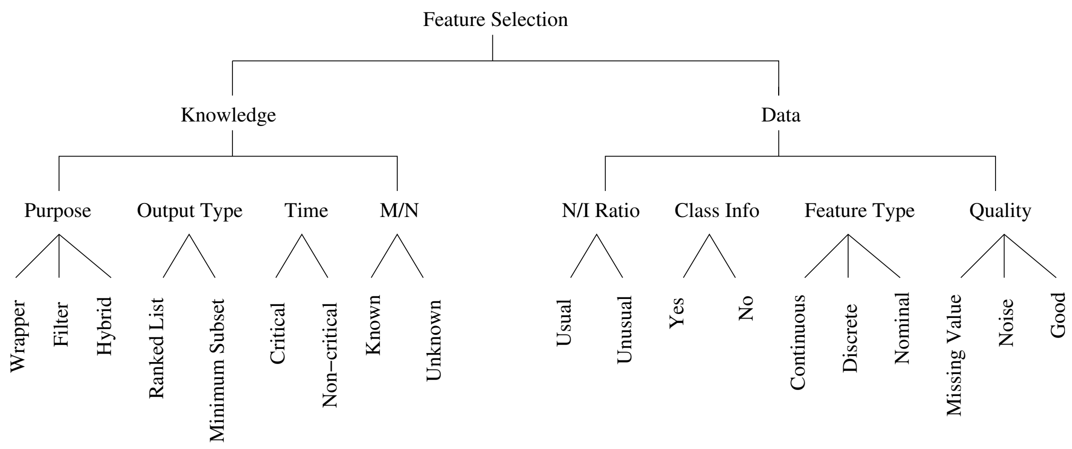

Figure 4: A Unifying Platform

ture selection are two key determining factors. Currently the knowledge factor covers why feature selection is conducted (or purpose), what the expected output type is (either a ranked list of features or a subset of features), whether it is time critical or not, and whether the number of selected features M is known w.r.t. $N(M/N)$. The data factor is concerned about data quality - noisy, value missing, or good, feature type - continuous, discrete, nominal, or mixed, class information - with/without class labels, and N/I ratio - the ratio between the number of features N and the number of instances I. Some dimensions are further discussed below.

The purpose of feature selection can be broadly categorized into visualization, understanding of data, data cleaning, redundancy and/or irrelevancy removal, and performance (e.g. prediction and comprehensibility) enhancement. Not every feature selection can serve all the purposes. As a first step, we summarize these different purposes in terms of filters or wrappers or a combination of the two to form a generic task hierarchy. For a wrapper model, many learning algorithms can be used: symbolic, connectionist, or semi-symbolic algorithms. A filter model can be used in applications where learning algorithms are too costly to employ, or unnecessary. Sometimes, we need to combine the filter and wrapper model into hybrid models to serve the more complicated purposes. This is important as different purposes guide the search of a feature selection method differently. The information on M/N is very useful in determining a proper search strategy. If M is known to be small, a forward complete search strategy can be afforded; if $N - M$ is small, a backward complete search strategy can be adopted even in time critical situations.

The N/I ratio becomes an interesting new problem lately when feature selection is applied to text mining, bio-informatics [XJK01]. In general, a large number of features poses much more difficulty to a feature selection algorithm than a huge number of instances does. The search space of a feature selection algorithm can be defined in terms of number of feature subsets, which is 2^N.

This exponential growth of the search space w.r.t the number of features (N) clearly rejects any attempts of brute force search when N is even moderately large (say,> 20). Usually, $I >> N$. Sometimes, however, N could be huge, but I small as in text mining. In such cases, we should focus on algorithms that intensively work along the I dimension. Different feature types require different data processing mechanisms. Some types of features inherit order in their values such as continuous and discrete features, some don't such as nominal features. When different features types occur in one data set, things become more complicated: how to consider each feature's influence. Mixed data types imply that the range of each feature values can vary significantly. It is important for our environment to recognize and allow this complexity for real-world applications. Data quality deals with missing values, noise, etc. Some applications require more preprocessing such as value discretization, missing value treatment, others are less stringent in this regard.

Domain knowledge, when available, should be used to aid feature selection. For example, a medical doctor may know that for a certain type of patient data, some features are more indicative than others, and some may be useless. This flexibility of using domain knowledge is not always required or necessary, especially in data mining application where we wish that the data would tell us this kind of information. But in the situations we can use this kind of knowledge, using it will certainly speed up the search. A particular interesting issue to us is how we can efficiently and succinctly represent this knowledge for feature selection.

4.4 Applying the platform on existing algorithms

An example presented in Figure 5 shows how the unifying platform can be used to categorize existing algorithms and what can be expected. There are 15 representative feature selection algorithms (B and B, WSFG, WSBG, SFG, SetCover, Relief, LVW, ABB, Focus, LVF, LVI, SBUD, Deva-Ram97, ELSA, Mitr-etal02) categorized in four aspects (class information, purpose, output type, search strategy which is influenced by time, and M/N information). The aspect of *Class Information* is about whether the data contains class information or not. With class information, supervised feature selection algorithms are needed, while without class information, unsupervised feature selection algorithms are needed. The aspect of *Purpose* is related to why feature selection is performed, which could suggest how selected features should be evaluated. *Output Type* of feature selection can sometimes be known *a priori*. This aspect divides feature selection algorithms into two groups: ranked lists and minimum subsets. The real difference between the two is about the order among the selected features. If a subset of features is chosen, then there is no order among the features. One cannot easily remove any more feature from the subset, but one can do so for a ranked list by removing the least important one. *Search strategy* is another important aspect that is determined by time constrain and knowledge about M - the expected number of selected features and N - the total number of original features. M may not be known and sometimes could be inferred from other sources of knowledge such as the purpose of feature selection, etc. Three

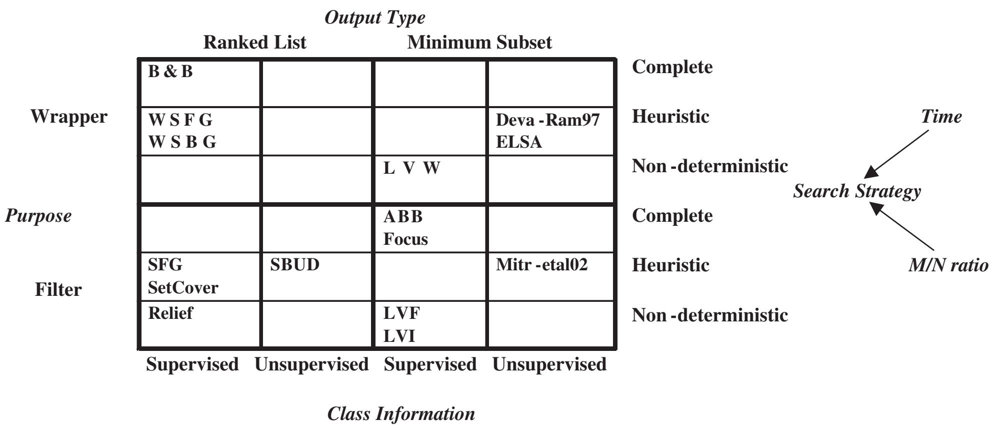

Figure 5: Applying the Unifying Platform

search strategies are Complete, Heuristic, and Non-deterministic.

5 Discussions and Future Work

In the future work, the work in Figure 3 will be substantially extended to include all the other factors in the unifying platform. Applying the unifying platform to existing algorithms serves two purposes: (1) to group existing algorithms to make sensible comparison, and (2) to find out what are missing. The immediate goal of the extensive comparison is three-fold:

1. investigating strengths and weaknesses of existing algorithms. We observe how these feature selection algorithms behave in the same unifying platform. Re-implementation of some existing algorithms is expected because we will have a more general environment.

2. uncovering how these algorithms complement each other. The complimentary natures of existing feature selection algorithms will be revealed in comparison. We can also study how we can serve all the specifiable needs to the maximum with the existing algorithms and how they can achieve best performances in varying special settings.

3. finding what's missing and what's needed. After we understand well the existing algorithms and their complimentary properties, we will be able to point out where we have accomplished sufficiently well and where we need to intensify our efforts. It is not necessary for us to create new algorithms for all the missing cells as in Figure 2. This is because that some combinations might be impossible to exist.

With the unifying platform, we wish to (1) discover the dominating algorithms under different circumstances; (2) uncover deficiencies in the current body of knowledge on feature selection that should be brought to light, so that subsequent work can be directed to resolve them; (3) identify untouched areas that allow for future expansion of the unifying platform; and (4) pave the way for the building of an automatic feature selection system.

References

[AD92] H. Almuallim and T.G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 547–552, Anaheim, California, 1992. AAAI Press/The MIT Press, Menlo Park, California.

[AD94] H. Almuallim and T.G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–305, November 1994.

[AIS93] R. Agrawal, T. Imielinski, and A. Swami. Database mining: A performance perspective. *IEEE Trans. on Knowledge and Data Engineering*, 5(6):914–925, December 1993.

[BB82] M. Ben-Bassat. Pattern recognition and reduction of dimensionality. In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of statistics-II*, pages 773–791. North Holland, 1982.

[BB96] G. Brassard and P. Bratley. *Fundamentals of Algorithms*. Prentice Hall, New Jersey, 1996.

[BL97] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[Bob88] L. Bobrowski. Feature selection based on some homogeneity coefficient. In *Proceedings of the Ninth International Conference on Pattern Recognition*, pages 544–546, 1988.

[BR92] A.L. Blum and R.L. Rivest. Training a 3-node neural networks is NP-complete. *Neural Networks*, 5:117 – 127, 1992.

[Car93] C. Cardie. Using decision trees to improve case-based learning. In P. Utgoff, editor, *Proceedings of the Tenth International Conference on Machine Learning*, pages 25–32, University of Massachusetts, Amherst, 1993.

[CF94] R. Caruana and D. Freitag. Greedy attribute selection. In *Proceedings of International Conference on Machine Learning (ICML-94)*, pages 28–36, Menlo Park, California, 1994. AAAI Press / The MIT Press.

[Das01] S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.

[DB00] J. G. Dy and C. E. Brodley. Feature subset selection and order identification for unsupervised learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 247–254, 2000.

[DK82] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall International, 1982.

[DL97] M. Dash and H. Liu. Feature selection methods for classifications. *Intelligent Data Analysis: An International Journal*, 1(3), 1997. `http://www-east.elsevier.com/ida/free.htm`.

[DL99] M. Dash and H. Liu. Handling large unsupervised data via dimensionality reduction. In *Proceedings of 1999 SIGMOD Research Issues in Data Mining and Knowledge Discovery (DMKD-99) Workshop*, 1999.

[DL00] M. Dash and H. Liu. Feature selection for clustering. In *Proceedings of Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining, (PAKDD-2000). Kyoto, Japan*. Springer-Verlag, 2000.

[DLM00] M. Dash, H. Liu, and H. Motoda. Consistency based feature selection. In *Proceedings of Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining, (PAKDD-2000). Kyoto, Japan*. Springer-Verlag, 2000.

[DLY97] M. Dash, H. Liu, and J. Yao. Dimensionality reduction of unsupervised data. In *Proceedings of the Ninth IEEE International Conference on Tools with AI (ICTAI'97), November, 1997*, pages 532–539, Newport Beach, California, 1997. IEEE Computer Society.

[Doa92] J. Doak. An evaluation of feature selection methods and their application to computer security. Technical report, Davis CA: University of California, Department of Computer Science, 1992.

[Dom97] P. Domingos. Why does bagging work? a Bayesian account and its implications. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 155 – 158. AAAI Press, 1997.

[DR97] M. Devaney and A. Ram. Efficient feature selection in conceptual clustering. In *Proceedings of Fourteenth International Conference on Machine Learning*, pages 92–97, 1997.

[FI92] U.M. Fayyad and K.B. Irani. The attribute selection problem in decision tree generation. In *AAAI-92, Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 104–110. AAAI Press/The MIT Press, 1992.

[Fis87] D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[FPSS96] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 495–515. AAAI Press / The MIT Press, 1996.

[FS87] I. Foroutan and J. Sklansky. Feature selection for automatic classification of non-gaussian data. *IEEE Transactions on Systems, Man, and Cybernatics*, SMC-17(2):187–198, 1987.

[GC85] M.A. Gluck and J.E Corter. Information, uncertainty, and the utility of categories. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, pages 283–87. Lawrence Erlbaum, Irvine, CA, 1985.

[Geo67] A. M. Geoffrion. Integer programming by implicit enumeration and balas's method. *SIAM Review*, 9(2):178–190, 1967.

[HF96] J. Han and Y. Fu. Attribute-oriented induction in data mining. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 399–421. AAAI Press / The MIT Press, 1996.

[IS84a] M. Ichino and J. Sklansky. Feature selection for linear classifier. In *Proceedings of the Seventh International Confernce on Pattern Recognition*, volume 1, pages 124–127, July-Aug 1984.

[IS84b] M. Ichino and J. Sklansky. Optimum feature selection by zero-one programming. *IEEE Trans. on Systems, Man and Cybernetics*, SMC-14(5):737–746, September/October 1984.

[J.78] Rissanen J. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

[JKP94] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant feature and the subset selection problem. In W.W. asnd Hirsh H. Cohen, editor, *Machine Learning: Proceedings of the Eleventh International Conference*, pages 121–129, New Brunswick, N.J., 1994. Rutgers University.

[JZ97] A. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

[KJ97] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[Kon94] I. Kononenko. Estimating attributes : Analysis and extension of RELIEF. In F. Bergadano and L. De Raedt, editors, *Proceedings of the European Conference on Machine Learning, April 6-8*, pages 171–182, Catania, Italy, 1994. Berlin: Springer-Verlag.

[KR92] K. Kira and L.A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 129–134. Menlo Park: AAAI Press/The MIT Press, 1992.

[KS96] D. Koller and M. Sahami. Toward optimal feature selection. In L. Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference (ICML-96), July 3-6, 1996*, pages 284–292, Bari, Italy, 1996. San Francisco: Morgan Kaufmann Publishers.

[KSM00] Y. Kim, W. Street, and F. Menczer. Feature selection for unsupervised learning via evolutionary search. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 365–369, 2000.

[Lan94] P. Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*. AAAI Press, 1994.

[LLFH99] H. Liu, H. Lu, L. Feng, and F. Hussain. Efficient search of reliable exceptions. In *Proceedings of the Third Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-99)*. Springer, 1999.

[LM98a] H. Liu and H. Motoda, editors. *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Boston: Kluwer Academic Publishers, 1998. 2nd Printing, 2001.

[LM98b] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery Data Mining*. Boston: Kluwer Academic Publishers, 1998.

[LM98c] H. Liu and H. Motoda. *Less Is More*, chapter 1, pages 3 – 12. In [LM98a], 1998. 2nd Printing, 2001.

[LS96a] H. Liu and R. Setiono. Feature selection and classification - a probabilistic wrapper approach. In T. Tanaka, S. Ohsuga, and M. Ali, editors, *Proceedings of the Ninth International Conference on Industrial and Engineering Applications of AI and ES*, pages 419–424, Fukuoka, Japan, 1996.

[LS96b] H. Liu and R. Setiono. A probabilistic approach to feature selection - a filter solution. In L. Saitta, editor, *Proceedings of International Conference on Machine Learning (ICML-96), July 3-6, 1996*, pages 319–327, Bari, Italy, 1996. San Francisco: Morgan Kaufmann Publishers, CA.

[LS98] H. Liu and R. Setiono. Scalable feature selection for large sized databases. In F. Cantu, R. Soto, J. Liebowitz, and E. Sucar, editors, *Proceedings of the Fourth World Congress on Expert Systems (WCES'98)*, Mexico City, Mexico, 1998. Morgan Kaufmann Publishers.

[MG71] A. N. Mucciardi and E. E. Gose. A comparison of seven techniques for choosing subsets of pattern recognition. *IEEE Transactions on Computers*, C-20:1023–1031, September 1971.

[ML94] A. W. Moore and M. S. Lee. Efficient algorithms for minimizing cross validation error. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 190–198, New Brunswick, New Jersy, 1994. Morgan Kaufmann.

[MMP02] P. Mitra, C. A. Murthy, and S. K. Pal. Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):301–312, 2002.

[Mod93] M. Modrzejewski. Feature selection using rough sets theory. In P.B. Brazdil, editor, *Proceedings of the European Conference on Machine Learning*, pages 213–226, Vienna, Austria, 1993.

[NF77] P.M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Trans. on Computer*, C-26(9):917–922, September 1977.

[OV92] A. L. Oliveira and A. S. Vincentelli. Constructive induction using a non-greedy strategy for feature selection. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 355–360, Aberdeen, Scotland, 1992. Morgan Kaufmann.

[PK99] F. Provost and V. Kolluri. A survey fo methods for scaling up inductive algorithms. *Journal of Data Mining and Knowledge Discovery*, 3:131 – 169, 1999.

[Pyl99] D. Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann Publishers, 1999.

[QG84] C. E. Queiros and E. S. Gelsema. On feature selection. In *Proceedings of the Seventh International Conference On Pattern Recognition*, pages 128–130, 1984.

[Qui93] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[Sch93] J. C. Schlimmer. Efficiently inducing determinations : a complete and systematic search algorithm that uses optimal pruning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 284–290, 1993.

[SDN90] J. Sheinvald, B. Dom, and W. Niblack. A modelling approach to feature selection. In *Proceedings of the Tenth International Conference on Pattern Recognition*, volume 1, pages 535–539, June 1990.

[Seg84] J. Segen. Feature selection and constructive inference. In *Proceedings of the Seventh International Conference on Pattern Recognition*, pages 1344–1346, 1984.

[Ska94] D. B. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 293–301, New Brunswick, NJ, 1994. Morgan Kaufmann.

[SS88] W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2:197–220, 1988.

[Tal99] L. Talavera. Feature selection as a preprocessing step for hierarchical clustering. In *Proceedings of Internationl Conference on Machine Learning (ICML'99)*, 1999.

[VI94] H. Vafaie and I. F. Imam. Feature selection methods: genetic algorithms vs. greedy-like search. In *Proceedings of the International Conference on Fuzzy and Intelligent Control Systems*, 1994.

[WDJ80] N. Wyse, R. Dubes, and A.K. Jain. A critical evaluation of intrinsic dimensionality algorithms. In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice*, pages 415–425. Morgan Kaufmann Publishers, Inc., 1980.

[WK91] S.M. Weiss and C.A. Kulikowski. *Computer Systems That Learn*. Morgan Kaufmann Publishers, San Mateo, California, 1991.

[XJK01] E. Xing, M. Jordan, and R. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference On Machine Learning*, 2001.

[XYC88] L. Xu, P. Yan, and T. Chang. Best first strategy for feature selection. In *Proceedings of the Ninth International Conference on Pattern Recognition*, pages 706–708, 1988.

[Zil96] S. Zilberstein. Using anytime algorithms in intelligent systems. *AI Magazine*, pages 73–83, Fall 1996.

	Evaluation Criteria	Search Strategies		
		Complete	Heuristic	Non-determininstic
Supervised	Distance Measure	Category 1 Branch and Bound BFF Bobr88	Category 2 Relief ReliefF Sege84	
	Information Measure	Category 3 MDLM	Category 4 SFG DTM Koll-Saha96	
	Dependency Measure		Category 5 POE+ACC RRESET	
	Consistency Measure	Category 6 Focus Schl93 MIFES1 B+B ABB		Category 7 LVF LVI QBB
	Classifier Error Rate	Category 8 Ichi-Skla84a Ichi-Skla84b AMB and B BS	Category 9 WSFG WSBG SBS-SLASH PQSS BDS Moor-Lee94 RC Quei-Gels84	Category 10 LVW GA SA RGSS RMHC-PF1
	Information+Accuracy		Category 11 BBHFS Xing-etal01	
Unsupervised	Information Measure		Category 12 SBUD	
	Similarity Measure		Category 13 Mitr-etal02	
	Cluster Goodness		Category 14 Deva-Ram97 FSSEM ELSA	
	Information + Cluster Goodness		Category 15 Dash-Liu00	